

## REMARKS

Applicants respectfully traverse and request reconsideration.

Amendments have been made in the Abstract, throughout the specification, in claims 1, 4, 10, and 13 and in Fig. 2 correcting various cosmetic errors. With respect to Fig. 2, a dotted line replaces a sold line from the PVS CNTRL (element 214) to the PVS ENGINE (element 202) indicating the manner by which the PVS engine receives control signals from he PVS controller. (Page 8, Lines 18-20). Applicants respectfully submit that no additional matter has been included in the aforementioned amendments.

Claims 1, 3-5, 9-13 and 15-16 stand rejected under 35 U.S.C. §102(e) as being anticipated by Kwok et al., U.S. Patent No. 6,088,044 ("Kwok"). Kwok is directed to a method for parallelizing and distributing work assignments in multiprocessor systems such as a graphics data processing system. (Col. 1, lines 15-20; Col 4, lines 1-6). More specifically, the system and method of Kwok teaches the use of multiple threads in a graphics pipeline of an image rendering system. (Column 3, lines 59-61). As illustrated in Fig. 6A and briefly described in the office action, the disclosed method taught by Kwok is for processing and storing *input data*, such as vertex and geometric input data, in the parallelization of a geometry pipeline. (Col. 10, lines 52-56 describing Fig. 6A; and Col. 4, lines 1-6; Col 11, lines 1-9; Fig. 6A. step 52 teaching the interchangeable use of the terms vertex data and geometric input data for term input data). While Kwok describes a method and apparatus for storing and processing input data in a multithread geometric pipeline (Fig. 6A), the method and apparatus is silent as to Applicants' claimed invention of storing *state data* in a buffer residing in the graphics processor.

In fact, the Kwok reference is explicit in distinguishing input data from state commands. Kwok defines input data as non-state commands that need to be processed (i.e., input data are operands to be processed based upon state commands). (Col. 11, lines 4-6). In contrast, state commands are defined by their function, i.e., they describe the overall state of the geometric processing operation, and may include a type of polygon, color(s), lighting direction, etc.

(Column 10, line 66 – column 11, line 2). State commands, according to Kwok, are utilized to process input data. While the reference discusses the existence and function of state commands, i.e., Kwok teaches that state commands do operate upon vertex data, the reference appears to be silent as to Applicants' claimed method of "receiving and storing N sets of state data in the buffer, where the total length of the N sets of state data does not exceed a length of the buffer...and prohibiting an additional set of state data from being stored in the buffer when N equals a maximum number of allowed states." In contrast, the cited portion of the reference appears to focus on the processing of vertex data in accordance with existing state command parameters. Moreover, Applicants respectfully note that the Office Action appears to improperly interchange input data and state commands. (See e.g., page 3, lines 13-14 of Office Action). As a result, Applicants respectfully believe that claim 1 is in proper condition for allowance.

With respect to claim 1, Applicants respectfully note that the office action appears to merely paraphrase the operation of Fig. 6A, a logic flow diagram illustrating the overall relationship between the main and child threads in addition to the flow of control in the parallelization of the geometry pipeline. While the summary of Fig. 6A appears to be accurate, the recited steps fail to disclose the method comprising "receiving and storing N sets of state data into the buffer, when the total length of the N sets of state data does not exceed a length of the buffer, and wherein at least one of the N sets of state data is used to process graphics primitives; and prohibiting an additional set of state data from being stored in the buffer when N equals a maximum number of allowed states."

Fig. 6A primarily deals with the input, storage and processing of vertex data (step 52) and merely mentions the manner by which state commands can affect the processing of vertex data. The reference teaches that when a child thread is created (steps 50), the main thread begins to accept geometric input data (i.e., vertex data) in step 52 and initiates filling a vertex buffer (Fig. 5, element 16) in steps 54-60. However, if a child thread does not need to be created

(because one already exists in step 48), a determination is made as to whether or not new state commands have arrived since the last vertex buffer was processed. Kwok teaches that if new state commands have arrived, then any unprocessed vertex data is first processed in accordance with the old state commands in steps 74-80 before the new state commands proceed using new parameters (e.g., color, viewpoint, lighting direction, etc.) (col. 11, Lines 20-36). No prohibition of storing state data appears to be discussed.

With respect to claims 5, 13 and 16, Applicants respectfully repeat the relevant remarks made above with respect to claim 1. More specifically, Applicants note that the office action has failed to anticipate a method for a graphics processor to store state data in a buffer residing in the graphics processor according to the limitations of claim 1. With respect to claim 13, Applicants respectfully submit that independent claim 13 parallels the subject matter claimed in claim 1 except that claim 13 is directed to an apparatus whereas claim 1 is directed to a method. Therefore, Applicants submit that claim 13 is allowable over Kwok for the same reasons set forth above with respect to claim 1. With respect to independent claim 5, Applicants respectfully resubmit the relevant remarks made in response to the previous office action. Specifically, Applicants note that independent method claim 5 is directed at a method for a host to update *state data* in a buffer residing in a graphics processor. More specifically, the method comprises “writing N sets of state data to the buffer, where the total length of the N sets of state data does not exceed a length of the buffer..., determining whether a length of an additional set of state data would exceed available space in the buffer; and when the length of the additional set of state data exceeds the available space in the buffer, waiting” until a requisite amount of buffer space is available and then writing the additional state data into the buffer. Because Kwok is primarily directed at a method for inputting and processing vertex data and fails to disclose a method by which length of state data are determined and are updated in a buffer or written to a buffer, Applicants respectfully submit that claim 5 is in proper condition for

allowance. With respect to independent claim 16, Applicants respectfully submit that independent claim 15 parallels the subject matter claimed in claim 5 except that claim 16 is directed to an apparatus whereas claim 5 is directed to a method. As such, Applicants submit that claim 16 is allowable over Kwok for the analogous reasons set forth above with respect to claim 5.

Referring to claims 3, 4, 10-13 and 15, Applicants respectfully repeat the relevant remarks made with regard to claims 1, 5, 13 and 16. The aforementioned claims contain further patentable subject matter and are in proper condition for allowance for the same or similar reasons as Claims 1, 5, 13 and 16. Applicants further note that with respect to claims 4 and 10, the Kwok reference is silent as to a method of storing or updating state data in a buffer wherein the buffer further comprises either a code buffer or a constant buffer. Applicants' specification teaches that state data may comprise either code data or constant data. (Page 4, lines 7-8). Furthermore, the specification teaches that in a preferred embodiment, code data is stored in a code buffer while constant data is stored in a constant buffer. (Page 4, lines 15-21). In contrast, Kwok merely describes that state commands are stored in a memory area that is accessible by the main thread and the child threads. (Column 11, lines 6-8). In fact, the Kwok reference does not appear to distinguish between code and constant state data. Applicants respectfully submit that the Kwok reference appears to be silent as to a method wherein the buffer comprises either a code buffer or a constant buffer.

In reference to claims 3, 9 and 15, Applicants respectfully note that the Office Action appears to focus on prohibiting *input data* from being stored in the Kwok system until the previous state commands have been executed (steps 70-80). However, Applicants reassert that the claimed invention focuses on storage of *state data* and not *input data*. For example, claim 3 recites a method wherein "the additional set of state data [is permitted] to be stored in the buffer when the M sets of state data are no longer being used to process the graphics primitives."

As state *infra*, Applicants respectfully believe claims 3, 4, 9-12 and 15 to be in proper condition for allowance.

Claims 2, 7, 8, 14, 18 and 19 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kwok et al. Applicants respectfully reassert the relevant remarks made above with respect to allowable claims 1, 5, 13 and 16. Specifically, Applicants note that Kwok does not appear to teach the method or apparatus whereby N sets of state buffer are received and stored in a buffer residing in the graphics process in the manner of claims 1 and 13. Nor does Kwok appear to teach the method or apparatus whereby N sets of state data are written to a buffer residing in the graphics process in the manner of claims 5 and 16. Instead the Kwok reference appears to focus on the storage and processing of input data in a multiprocessor system. Because claims 2, 7, 8 and 14 rely upon allowable base claims and further contain nonobvious patentable subject matter over Kwok, Applicants respectfully believe claims 2, 7, 8 and 14 are allowable.

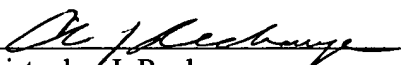
Claims 6 and 17 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Kwok et al in view of alleged admitted prior art. Applicants respectfully repeat the relevant remarks made with respect to allowable claims 5 and 16 and note that claims 6 and 17 contain further patentable and nonobvious subject matter of Kwok and the admitted prior art. Moreover, Applicants respectfully note that while the specification teaches that ring buffers are well known to those having ordinary skill in the art, Applicants do not admit that ring buffers are commonly used in this specific context of updating state data. Nor do Applicants admit that one of ordinary skill in the art would utilize a ring buffer to solve delay problems caused by traditional state update systems. In fact, Applicants submit that the basis of utilizing Applicants' specification is little more than impermissible hindsight analysis. Per MPEP § 2142, the conclusion of obviousness must be reached on the basis of the facts gleaned from the prior art putting aside applicant's disclosure. In this case, the only apparent source of the teaching of utilizing a ring

buffer in connection with Applicants' claimed invention is the instant application. As such, Applicants respectfully submit that the rejection of claims 6 and 17 under § 103(a) improperly engages in hindsight analysis and therefore fails to establish a proper basis of the rejection of the aforementioned claims.

Accordingly, Applicants respectfully submit that the claims are in condition for allowance and that a timely Notice of Allowance be issued in this case. The Examiner is invited to contact the below-listed attorney if the Examiner believes that a telephone conference will advance the prosecution of this application.

Respectfully submitted,

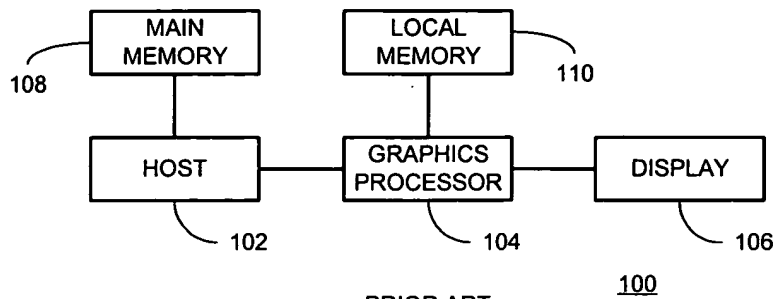
Dated: April 29, 2004

By:   
Christopher J. Reckamp  
Reg. No. 34,414

Vedder, Price, Kaufman & Kammholz, P.C.  
222 North LaSalle  
Chicago, Illinois 60601-1003  
312/609-7500  
312/609-5005 Facsimile



Annotated Sheet Showing Changes



- PRIOR ART -

FIG. 1

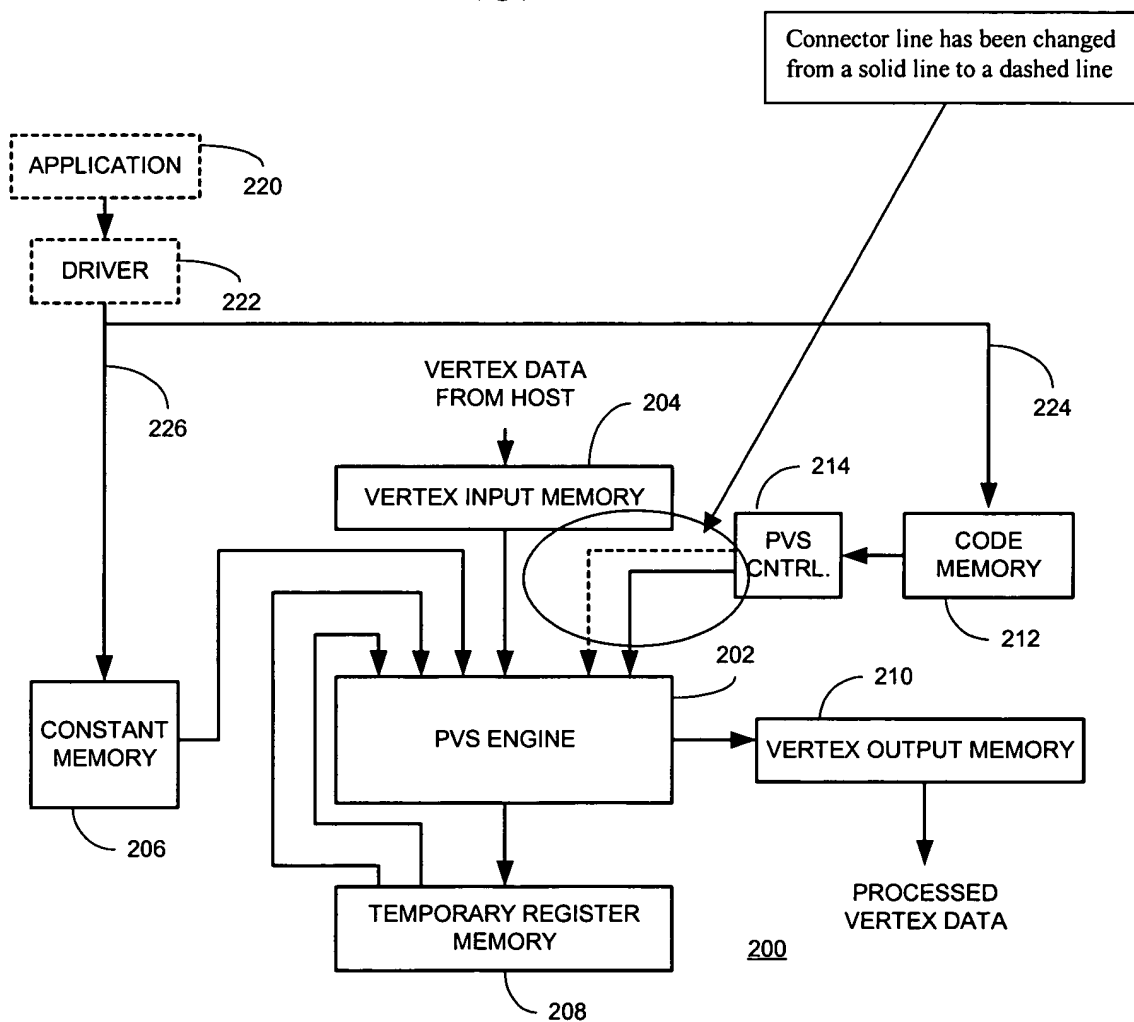


FIG. 2